



# Microsoft

## 70-489 Exam

### Developing Microsoft SharePoint Server 2013 Advanced Solutions

Thank you for Downloading 70-489 exam PDF Demo

You can Buy Latest 70-489 Full Version Download

<https://www.certkillers.net/Exam/70-489>

<https://www.certkillers.net>

**Case Study: 1****Trey Research****Background**

You develop an intranet portal for Trey Research. End users of the portal are researchers and office staff.

**Business Requirements**

All end users must be able to customize their profile with relevant information. Researchers must store research papers, upload supporting documents, and search content.

**Storage**

The portal must use an existing Microsoft SQL Server database to access and store work profile information and research papers.

**Data Access**

The portal must use Business Connectivity Services (BCS) to access data from external systems.

Researchers must search content from SharePoint and external systems.

Researchers must manage a research topic and related content as a single entity.

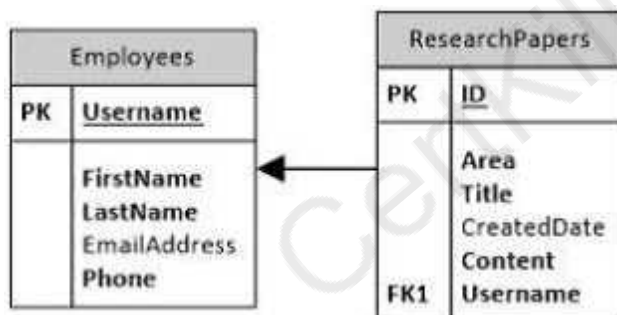
**User Profile**

Employees must be able to customize their profile.

Administrators must be able to create new profile properties.

**Technical Requirements****Data Store**

The data model for the database entities is shown below:



Users must not be allowed to update the Employees.Username and ResearchPapers.ID fields. The fields uniquely distinguish the corresponding entity.

**Access External Data**

You must create an external content type named TreyResearch to access the SQL data source. During development, the data source will be accessible locally.

You must develop an app to access the fields named Employee Name and Research Paper Title.

Researchers must be able to find all research papers that are written by a particular employee.

A research paper always must be associated with the employee that wrote it.

**Document Management**

Researchers must be able to upload research papers and relevant supporting materials into a document set named Research Content.

All the document sets must be stored in a list named ResearchPapers.

All documents that are uploaded must contain the prefix DOC in the file name.

**Environment**

The SQL database will be on a different physical server when the solution is deployed to a production environment. The solution must use the SQL Server user named sqltrey to connect to the database. The BCS service is configured and running in the production environment.

**Personalize**

You must use custom profile properties to add a new section to the user profile properties page.

The solution must use the client-side object model (CSOM) to upload employee profile pictures.

Employees must be able to change their display name on the site.

Each employee's page must display the value of the DisplayName and Title fields.

**Search**

The Microsoft Bing API web service must be used to search for research papers.

No code must be written.

The app must use a Content Enrichment web service named AbstractIndexer. The app must use the AbstractIndexer service to index search content.

The solution must store large-sized media files in a dedicated SQL Server database.

The database must use the ResearchPapers.ID field as the foreign key to associate the field with the TreyResearch external content type.

**Application Structure**

Relevant portions of the solution files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

**App.js**

```
AJ01  var context;
AJ02  var web;
AJ03  var user;
AJ04
AJ05
AJ06  $.ajax({
AJ07      url: listURL,
AJ08      headers: {
AJ09          "accept": "application/json",
AJ10          "X-RequestDigest": $("#__REQUESTDIGEST").val()
AJ11      },
AJ12      success: this.showItems,
AJ13      error: this.failMethod
AJ14  });
AJ15  }
AJ16
AJ17  this.showItems = function (data) {
AJ18      $("#Container").children().remove();
AJ19      $.each(data.d.results, function (key, val) {
AJ20          var item = $("#EmployeeInfoTemplate").clone()
AJ21          .attr("id", val.BdcIdentity)
AJ22          .fadeIn("slow");
AJ23          ...
AJ24          item.appendTo("#Container");
AJ25      });
AJ26  }
AJ27
AJ28  this.failMethod = function (jqXHR, textStatus, errorThrown) {
AJ29      alert('failed: ' + errorThrown);
AJ30  }
AJ31  }
AJ32  ExecuteOrDelayUntilScriptLoaded(getEmployees, "sp.js");
AJ33  });
AJ34
AJ35  function getEmployees() {
AJ36      var grid = new AppLevelECT.Grid
AJ37      ("ColumnContainer", 3, _spPageContextInfo.webServerRelativeUrl);
AJ38      grid.init();
AJ39  }
```

**ManageUserProfiles.es**

```

MP01 namespace ManageUserProfiles
MP02 {
MP03     class ProfileProperties
MP04     {
MP05         public static void AddProfileProperty(string name, string displayName,
MP06         bool isMultivalued)
MP07         {
MP08             using (SPSite site = new SPSite("http://treyresearch.com/users"))
MP09             {
MP10                 SPServiceContext svcContext = SPServiceContext.GetContext(site);
MP11                 try
MP12                 {
MP13                     ProfilePropertyManager prfPropMgr;
MP14                     ProfileSubtypeManager prfTypeMgr;
MP15                     ProfileSubtypePropertyManager prftypePropMgr;
MP16                     ProfileTypePropertyManager typPropMgr;
MP17                     ProfileSubtypeProperty prfTypeProp;
MP18                     ProfileTypeProperty prfProp;
MP19                     ProfileSubtype prfType;
MP20                     CorePropertyManager corePropMgr;
MP21                     CoreProperty coreProp;
MP22                     prfPropMgr = new UserProfileConfigManager(svcContext)
MP23                     .ProfilePropertyManager;
MP24
MP25                     prfTypeProp = prftypePropMgr.Create(prfProp);
MP26                     prfTypeProp.IsUserEditable = true;
MP27                     prfTypeProp.DefaultPrivacy = Privacy.Public;
MP28                     prfTypeProp.UserOverridePrivacy = true;
MP29                     prftypePropMgr.Add(prfTypeProp);
MP30                 }
MP31                 catch (System.Exception e)
MP32                 {
MP33                     throw new Exception("Error occurred: " + e.ToString());
MP34                 }
MP35             }
MP36         }
MP37     }
MP38 }
MP39
MP40 public void UploadPicture(string account, string picURL)
MP41 {
MP42     try
MP43     {
MP44     }
MP45     catch (Exception e)
MP46     {
MP47         throw new Exception("Error occurred: " + e.ToString());
MP48     }
MP49 }
MP50 }
MP51
MP52 public UserProfileProperties GetUserProfileProperties(string account)
MP53 {
MP54     var userprfProps = new UserProfileProperties();
MP55
MP56     var clientContext = new ClientContext("http://treyresearch.com/users");

```

ContentManagement.es

```

CM01 private void CreateDocumentSets()
CM02 {
CM03     using (SPSite site = new SPSite("http://treyresearch.com/sites"))
CM04     {
CM05         using (SPWeb web = site.RootWeb)
CM06         {
CM07
CM08         }
CM09     }
CM10 }

```

### Question: 1

DRAG DROP

You need to add code to line MP22 to create the custom profile property.

How should you complete the relevant code? (To answer, drag the appropriate code segments to the correct locations in the answer area. Each code segment may be used once or not at all. You may need to drag the split bar between panes or scroll to view content.)

	Answer Area
Create(coreProp)	corePropMgr = prfPropMgr.GetCoreProperties();
Create(corePropMgr)	coreProp = corePropMgr. <input type="text"/> ;
Create(true)	coreProp.Name = name;
Create(false)	coreProp.DisplayName = displayName;
GetProfileSubtypeProperties	coreProp.IsMultivalued = isMultivalued;
GetProfileTypeProperties	coreProp.Type = PropertyDataType.StringMultiValue;
GetCoreProperties	coreProp.Length = 1024;
	corePropMgr.Add(coreProp);
	typPropMgr = prfPropMgr
	. <input type="text"/> (ProfileType.User);
	prfProp = typPropMgr. <input type="text"/> ;
	prfProp.IsVisibleOnViewer = true;
	typPropMgr.Add(prfProp);
	prfTypePropMgr = prfPropMgr
	. <input type="text"/> (prfType.Name);

**Answer:**



**Available Code Segments:**

- Create(coreProp)
- Create(corePropMgr)
- Create(true)
- Create(false)
- GetProfileSubtypeProperties
- GetProfileTypeProperties
- GetCoreProperties

**Answer Area Code:**

```
corePropMgr = prfPropMgr.GetCoreProperties();
coreProp = corePropMgr. Create(true) ;

coreProp.Name = name;
coreProp.DisplayName = displayName;
coreProp.IsMultivalued = isMultivalued;
coreProp.Type = PropertyDataType.StringMultiValue;
coreProp.Length = 1024;
corePropMgr.Add(coreProp);
typPropMgr = prfPropMgr

. GetProfileTypeProperties (ProfileType.User);

prfProp = typPropMgr. Create(coreProp) ;

prfProp.IsVisibleOnViewer = true;
typPropMgr.Add(prfProp);
prfTypePropMgr = prfPropMgr

. GetProfileSubtypeProperties (prfType.Name);
```

## Question: 2

DRAG DROP

You need to add code to line MP57 to display the required properties for the user profile. How should you complete the relevant code? (To answer, drag the appropriate code segments to the correct locations in the answer area. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)

**Available Code Segments:**

- clientContext.Load(userPrfProps);
- clientContext.LoadQuery(userPrfProps);
- .GetUserProfilePropertyFor(userPrfProps);
- .GetUserProfilePropertiesFor(userPrfProps);
- string[] prfProps = new string[] { "DisplayName", "Title" };
- string[] prfProps = new string[] { "PreferredName", "Title" };
- var prfProps = new AsyncList() { "DisplayName", "Title" };

**Answer Area Code:**

```
var peopleManager = new PeopleManager(clientContext);

var userPrfProps = new UserProfilePropertiesForUser(clientContext,
account, prfProps);
var profilePropertyValues = peopleManager.

clientContext.ExecuteQuery();
```

**Answer:**

**Available Code Segments:**

- clientContext.Load(userPrfProps);
- clientContext.LoadQuery(userPrfProps);
- .GetUserProfilePropertyFor(userPrfProps);
- .GetUserProfilePropertiesFor(userPrfProps);
- string[] prfProps = new string[] { "DisplayName", "Title" };
- string[] prfProps = new string[] { "PreferredName", "Title" };
- var prfProps = new AsyncList() { "DisplayName", "Title" };

**Answer Area Code:**

```
var peopleManager = new PeopleManager(clientContext);

string[] prfProps = new string[] { "DisplayName", "Title" };

var userPrfProps = new UserProfilePropertiesForUser(clientContext,
account, prfProps);
var profilePropertyValues = peopleManager.
.GetUserProfilePropertiesFor(userPrfProps);

clientContext.Load(userPrfProps);

clientContext.ExecuteQuery();
```

## Question: 3

You need to configure the external content type to search for research papers.

Which indexing connector should you use?

- A. .NET Type Connector
- B. WCF Service Connector
- C. Custom Connector
- D. SQL Server Connector

---

**Answer: B**

---

---

**Question: 4**

---

You need to generate document identifiers for each new document that is uploaded to the site. What should you do?

- A. Create a derived class that inherits from the abstract class named Microsoft.Office.DocumentManagement.DocumentId and then override all of the abstract methods.
- B. Create a derived class that inherits from the abstract class named Microsoft.Office.DocumentManagement.DocumentIdProvider and then override all of the virtual members.
- C. Create a derived class that inherits from the Microsoft.Office.DocumentManagement.DocumentIdProvider abstract class and then implement all abstract members.
- D. Create a class to implement the Microsoft.Office.DocumentManagement.IDocumentId interface and then override all of the virtual members.

---

**Answer: B**

---

---

**Question: 5**

---

DRAG DROP

You need to configure authentication for the external content type in the production environment. Which three actions should you perform in sequence? (To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.)



	Answer Area
Add the sqitrey user login to the Secure Store Service.	
Configure permissions to allow the user that is logged in to have access to the Secure Store Service.	
Stop the Business Data Connectivity service.	
Create a Secure Store Service application as a target application.	
Connect to the external data source by using the Impersonated Custom Identity and the target application name.	
Connect to the external data source by using the Impersonated Windows Identity and the target application name.	

---

**Answer:**

---

**Answer Area**

Add the sqltre user login to the Secure Store Service.

Configure permissions to allow the user that is logged in to have access to the Secure Store Service.

Stop the Business Data Connectivity service.

Create a Secure Store Service application as a target application.

Connect to the external data source by using the Impersonated Custom Identity and the target application name.

Connect to the external data source by using the Impersonated Windows Identity and the target application name.

Stop the Business Data Connectivity service.

Add the sqltre user login to the Secure Store Service.

Connect to the external data source by using the Impersonated Windows Identity and the target application name.

### Question: 6

#### DRAG DROP

You need to add code at line AJ05 to complete the implementation of the app.

How should you complete the relevant code? (To answer, drag the appropriate code segment to the correct location in the answer area. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)

**Answer Area**

context.load(user);

context.init(user);

Lists/GetByTitle

Lists/GetByName

FirstName, LastName, Title

FirstName, LastName, Phone

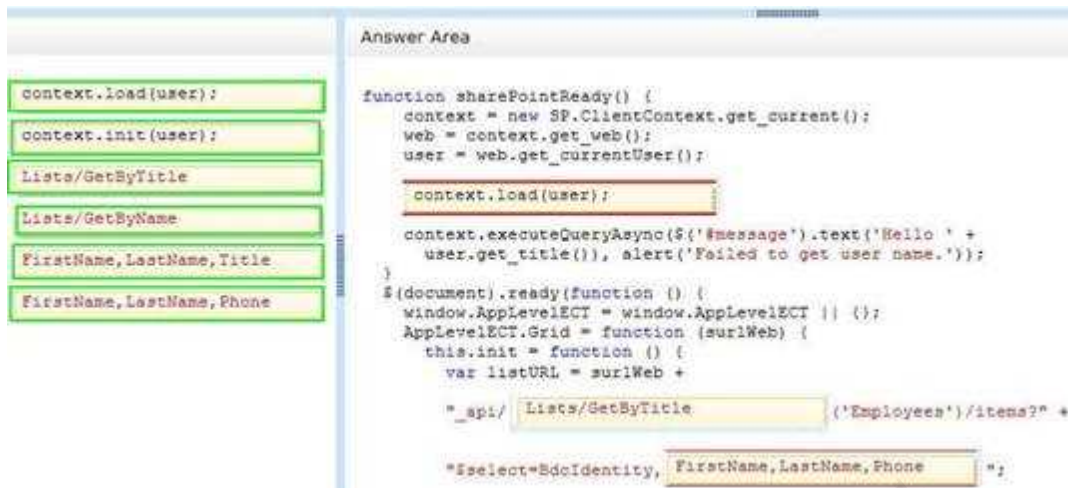
```
function sharePointReady() {
  context = new SP.ClientContext.get_current();
  web = context.get_web();
  user = web.get_currentUser();

  context.executeQueryAsync($('%message').text('Hello ' +
    user.get_title()), alert('Failed to get user name.'));
}
$(document).ready(function () {
  window.AppLevelECT = window.AppLevelECT || {};
  AppLevelECT.Grid = function (urlWeb) {
    this.init = function () {
      var listURL = urlWeb +
        "_api/" + ('Employees') + "/items?" +
        "$select=BdcIdentity, ";
```

---

**Answer:**

---



The screenshot shows the 'Answer Area' of a SharePoint application. On the left, a list of options is displayed, with the following items selected (highlighted in green):

- context.load(user);
- context.init(user);
- Lists/GetByTitle
- Lists/GetByName
- FirstName, LastName, Title
- FirstName, LastName, Phone

On the right, the JavaScript code is shown, with the selected options inserted into the code:

```
function sharePointReady() {
    context = new SP.ClientContext.get_current();
    web = context.get_web();
    user = web.get_currentUser();

    context.load(user);

    context.executeQueryAsync($('#message').text('Hello ' +
        user.get_title()), alert('Failed to get user name.));
}

$(document).ready(function () {
    window.AppLevelECT = window.AppLevelECT || {};
    AppLevelECT.Grid = function (surlWeb) {
        this.init = function () {
            var listURL = surlWeb +

            "_api/ Lists/GetByTitle ('Employees')/items?" +

            "$select=Id,Identities, FirstName, LastName, Phone ";
        }
    }
}
```

---

**Question: 7**

---

**HOTSPOT**

You need to create the external content type to support the data model.

In the Return Parameter Configuration dialog box, in the properties section, which options should you select? (To answer, select the appropriate setting or settings in the answer area.)

Properties	
Data Source Element:	Username
.NET Type:	System.String
Map to Identifier:	<input checked="" type="checkbox"/>
Identifier:	<input type="text" value="Username"/>
Field:	<input type="text" value="Username"/>
Display Name:	<input type="text" value="Username"/>
Foreign Identifier:	<a href="#">(Click to Add)</a>
Required:	<input type="checkbox"/>
Read-Only:	<input type="checkbox"/>
Office Property:	<input type="text" value="Unmapped"/>
Show In Picker:	<input type="checkbox"/>
Timestamp Field:	<input type="checkbox"/>

---

**Answer:**

---

Properties	
Data Source Element:	Username
.NET Type:	System.String
Map to Identifier:	<input checked="" type="checkbox"/>
Identifier:	Username
Field:	Username
Display Name:	Username
Foreign Identifier:	(Click to Add)
Required:	<input type="checkbox"/>
Read-Only:	<input type="checkbox"/>
Office Property:	Unmapped
Show In Picker:	<input type="checkbox"/>
Timestamp Field:	<input type="checkbox"/>

---

**Question: 8**

---

HOTSPOT

You need to add code to line CM07 to create the Research Content document set.

How should you complete the relevant code? (To answer, select the appropriate option from the dropdown list in the answer area.)

```

SPList lst = web.Lists[""];

SPFolder folder = lst.RootFolder;
SPContentType docsetCT = lst.ContentTypes["Document Set"];

Hashtable properties = new ();

properties.Add("Name", "Research Name");
properties.Add("Description", "Research Description");
properties.Add("Content1", "Video File Name");
properties.Add("Content2", "Audio File Name");
var docSet = DocumentSet.

;

SPList lst = web.Lists[""];
RootFolder
Document Sets
ResearchPapers
Shared Documents

SPFolder folder = lst.RootFolder;
SPContentType docsetCT = lst.ContentTypes["Document Set"];

Hashtable properties = new ();
ArrayList
Dictionary
Hashtable

properties.Add("Name", "Research Name");
properties.Add("Description", "Research Description");
properties.Add("Content1", "Video File Name");
properties.Add("Content2", "Audio File Name");
var docSet = DocumentSet.

;
Create(folder, "Research Content", docsetCT.Id, properties);
Create(folder, "Research Content", lst, properties);
Create(folder, "Research Documents", docsetCT.Id, properties, true);


```

---


**Answer:**



```

SPList lst = web.Lists[" ();

properties.Add("Name", "Research Name");
properties.Add("Description", "Research Description");
properties.Add("Content1", "Video File Name");
properties.Add("Content2", "Audio File Name");
var docSet = DocumentSet.


Create(folder, "Research Content", docsetCT.Id, properties);
Create(folder, "Research Content", lst, properties);
Create(folder, "Research Documents", docsetCT.Id, properties, true);

```

---

### Question: 9

---

You need to configure authentication to access the SQL data source during development. Which authentication mechanism should you use?

- A. Impersonated Windows Identity
- B. Pass Through
- C. Impersonated Custom Identity
- D. Forms Based Authentication

---

**Answer: B**

---



---

### Question: 10

---

You need to ensure that users can upload pictures. Which code segment should you insert at line MP57?

- ☐ A. 

```
using (SPSite site = new SPSite("http://treyresearch.com/users"))
{
    var upm = new UserProfileManager(clientContext);
    var up = upm.GetUserProfile(account);
    up["PictureUrl"].Value = picURL;
    up.Commit();
}
```
- ☐ B. 

```
var peopleManager = new PeopleManager(clientContext);
var personProperties = peopleManager.GetPropertiesFor(account);
...
Stream sr = new System.IO.FileStream(picURL, FileMode.Open);
peopleManager.SetMyProfilePicture(sr);
...
```
- ☐ C. 

```
using (SPSite site = new SPSite("http://treyresearch.com/users"))
{
    var upm = new UserProfileManager(clientContext);
    var up = upm.GetUserProfile(account);
    Stream sr = new System.IO.FileStream(picURL, FileMode.Open);
    up.PictureUrl.SetMyProfilePicture(sr);
    up.Commit();
}
```
- ☐ D. 

```
var peopleManager = new PeopleManager(clientContext);
var personProperties = peopleManager.GetPropertiesFor(account);
...
Stream sr = new System.IO.FileStream(picURL, FileMode.Open);
personProperties.PictureUrl = picURL;
...
```

- A. Option A  
B. Option B  
C. Option C  
D. Option D

---

**Answer: B**

---

## Thank You for trying 70-489 PDF Demo

To Buy Latest 70-489 Full Version Download visit link below

<https://www.certkillers.net/Exam/70-489>

## Start Your 70-489 Preparation

[Limited Time Offer] Use Coupon “CKNET” for Further discount on your purchase. Test your 70-489 preparation with actual exam questions.

<https://www.certkillers.net>